StratiGraph User's Guide

Pedher Johansson

UMINF 03.21
ISSN-0348-0542

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 And so it begins . . . 1
- 1.3 Outline . . . 3
- 1.4 Collaboration . . . 3

2 Some definitions **5**
- 2.1 Matrices and Jordan structures . . . 5
- 2.2 Matrix pencils and Kronecker structures . . . 6
- 2.3 Matrix pairs . . . 7
- 2.4 GUPTRI form . . . 7
- 2.5 Orbits and bundles . . . 8
- 2.6 Codimension . . . 8
- 2.7 Integer partitions . . . 9
- 2.8 Stratification . . . 9

3 Stratification by example **11**
- 3.1 A nilpotent matrix . . . 11
- 3.2 Canonical forms and Weyr characteristics . . . 11
- 3.3 Covered structures . . . 12
- 3.4 Covering structures . . . 13
- 3.5 Canonical structure hierarchy . . . 14

4 Describe your problem setup **17**
- 4.1 Supported problem setups . . . 17
- 4.2 Fixed or nonfixed eigenvalues . . . 17
- 4.3 Problem size . . . 19

5 Expanding a graph **21**
- 5.1 The node . . . 21
- 5.2 Expansion . . . 21

5.3 The relation between neighbouring nodes . . . 23
5.4 Rearrange a graph . . . 23
5.5 Structure overview in separate windows . . . 24
5.6 Scaling and overview of the graph . . . 24

6 Additional functionality **27**
6.1 Exporting and printing a graph . . . 27
6.2 Saving and loading a graph . . . 28
6.3 Different notation and font sizes . . . 28
6.4 Viewing options . . . 29
6.5 Loading Plug-ins . . . 30

7 Limitations and future developments **31**
7.1 Current limitations . . . 31
7.2 Supported problem setups . . . 31
7.3 Expansion options . . . 31
7.4 Quantitative results . . . 32

Appendix A Examples **33**
Example 1 – Nilpotent matrix . . . 33
Example 2 – Matrix as orbit and bundle . . . 33
Example 3 – Matrix pencils . . . 35
Example 4 – Control application . . . 37

Appendix B StratiGraph commands **41**
Menu bar . . . 41
Button bar . . . 43

Appendix C Keyboard short cuts **45**
Global short cuts . . . 45
Graph window short cuts . . . 45

Bibliography **47**

Index **49**

List of Figures

1 StratiGraph manin window 2

2 Relation between some canonical structures of a 6×6 nilpotent matrix orbit. The top three nodes in the graph correspond to the similarity orbits of (3.7), (3.6) and (3.1), respectively. 15
3 Jordan structure hierarchy of a 6×6 nilpotent matrix orbit. 15

4 The dialog window shown when StratiGraph is started. 17
5 The first step of the new graph wizard where the problem setup type is specified. 18
6 The second step of the new graph wizard where the choice between orbits and bundles are made. 18
7 The third step of the new graph wizard where the choice of starting point and size is made, illustrating the choice to start with the most generic matrix pencil of size 2×3. 20
8 The third step of the new graph wizard, illustrating the choice to start with a matrix pencil with a specific KCF. 20

9 Illustration and description of the nodes. 21
10 Node decorations 22
11 Expanding the graph upwards and downwards from the active node. 22
12 Dialog window shown when recursive expansion is done. 23
13 Illustration and description of the edges. 23
14 Window showing covering, active and covered structures. 24
15 Window showing all generated structures. 25
16 Illustration of how the graph scale is set. 25

17 The dialog window for exporting a graph with corresponding structure information to PostScript. 27
18 The same structure information shown with different notations. 29
19 Dialog window used to adjust horizontal and vertical distances between nodes. 29
20 The plug-in manager window. 30

21 The graph wizards illustrating how to specify a nilpotent matrix of size 6×6. 34
22 The orbit stratification of a 6×6 nilpotent matrix. 34
23 The orbit stratification of a 4×4 matrix. 36
24 The bundle stratification of a 4×4 matrix. 36
25 The graph wizards illustrating how to specify a matrix pencil of size 3×5. . 37
26 The complete graph illustrating the stratification of a 3×5 matrix pencil. . . 38
27 Illustration of the subgraph that a matrix pair forms in the matrix pencil graph. 40

CHAPTER 1

Introduction

1.1 Background

The determination of the Jordan form of a matrix or the Kronecker form of a matrix pair or pencil is an ill-posed problem in the presence of roundoff errors. Therefore there exists modern numerical software, such as GUPTRI [2, 3] that regularizes these problems by allowing a tolerance for rank decisions to find their canonical structure. However, the algorithms used are known to occasionally fail and thereby accidentally producing wrong, but nearby structures. Failure appears to occur when the matrix or pencil is close to a manifold of interesting structures of higher codimension. Alan Edelman, Erik Elmroth and Bo Kågström [4, 5] have proposed to make use of the mathematical knowledge of *stratification* of the Jordan and Kronecker structures in order to enhance the staircase algorithm. The stratification, in effect, shows which structures are nearby other structures (in the sense of being in the closure) in the space of matrices. The stratification can be described as a connected graph, that grows exponentially with increasing matrix dimension.

StratiGraph is a Java-based tool that gives a unique opportunity to view and investigate qualitative information on the relation between different canonical structures of an input setup. The stratification shows how the canonical structure can change for small perturbations in the input data. Together with software for quantitiative analysis this gives a very good understanding on many aspects of an input problem.

StratiGraph version 1.0 was initially prototyped within a Master thesis project and has been further developed into the new version 1.4. Since the original version, the software tool has gained a lot of functionality to help the user understanding the nature of his/her input problem.

1.2 And so it begins

In Figure 1, parts of the StratiGraph environment is shown. In the main window a graph representing the stratification of a matrix pencil is displayed. Here the complete graph is expanded but more common is to view a closure relation between a subset of different canonical structures. On the right, a window which lists all the expanded canonical structures is

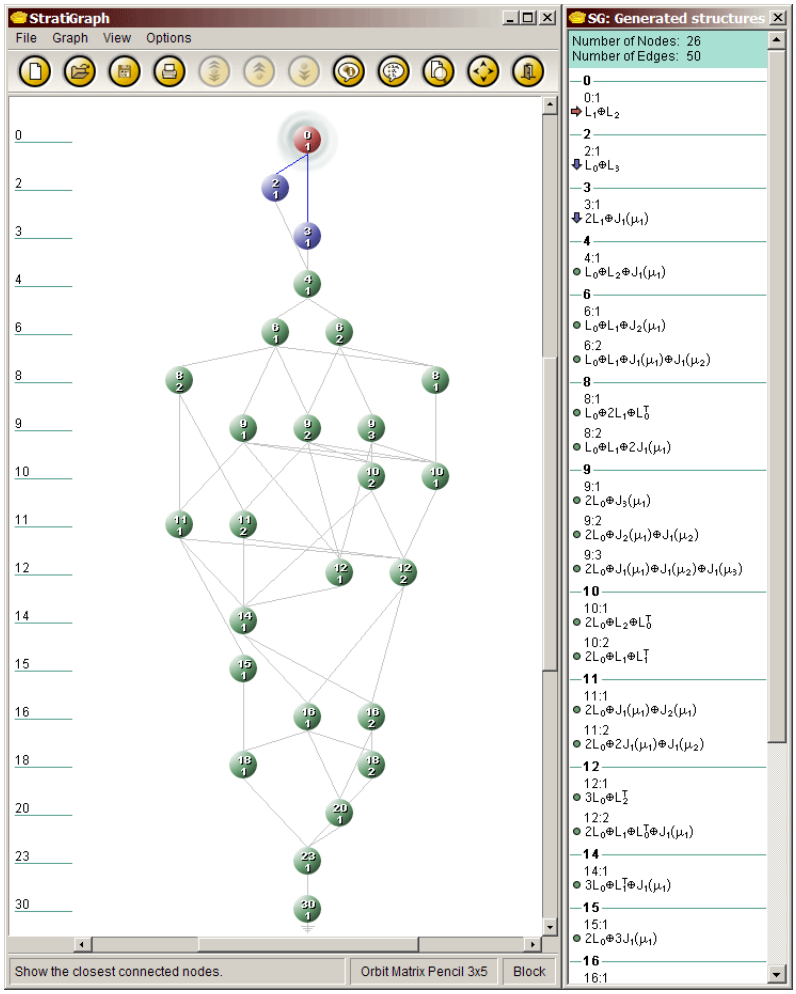

FIGURE 1: In the StratiGraph main window (left), an example of a connected graph representing the stratification of a 3×5 matrix pencil is shown. On the right is a list of all the canonical structures visible in a separate window.

shown.

In this User's Guide all the functionalities of the software will be covered, including the background information nescessary for understanding and make use of the information presented.

1.3 Outline

Chapter 2 presents some definitions and explanations of the terminology used in this document. The Jordan and Kronecker canonical forms and the meaning of orbits and bundles are explained and finally there is a discussion on integer partitions and their application in the stratification theory.

In Chapter 3, we present an example illustrating the use of the stratification in practice. Covering relations and different notations are disscussed. It is also shown, how small perturbations in the input problem can result in different canonical structures.

Chapter 4 explains how different input setups (matrix, matrix pencil or matrix pair) can be given to StratiGraph. Additional parameters in the setup specify how eigenvalues should be treated, how to specify the problem size, and which canonical structure to start from.

The expansion of a graph and the relation between different nodes are discussed in Chapter 5. How to get the best overview of available information and the graph is also covered.

Additional functionalities of StratiGraph are presented in Chapter 6. Chapter 7 and discuss current limitations, ongoing work and research. In Appendix A, a set of different examples are given. Appendix B covers all available commands in StratiGraph and Appendix C covers the available short cuts.

1.4 Collaboration

This work is done in close collaboration with professor Bo Kågström[1] and associate professor Erik Elmroth[2]. Stefan Johansson[3] has also helped a lot with the research, testing of the software and finding bugs.

[1]Umeå University, bokg@cs.umu.se
[2]Umeå University, elmroth@cs.umu.se
[3]Umeå University, stefanj@cs.umu.se

CHAPTER 2

Some definitions

2.1 Matrices and Jordan structures

An $n \times n$ matrix A with n distinct eigenvalues with corresponding eigenvectors x_i, has the following decomposition:

$$AX = XJ,$$

where $X = [x_1, x_2, ..., x_n]$ and $J = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$. Matrix A is then said to be *diagonalizable*.

A matrix with multiple eigenvalues can have less than n linearly independent eigenvectors. Such a matrix is not diagonalizable, and can not be decomposed into the form above. However, a matrix can always be transformed into a block diagonal form [7]:

$$X^{-1}AX = \text{diag}(\lambda_1 I + N_1, ..., \lambda_q I + N_q),$$

where $\lambda_1, ..., \lambda_q$ are distinct eigenvalues and N_i is a strictly upper triangular nilpotent matrix of size $a_i \times a_i$. The size a_i is called the *algebraic multiplicity* of the eigenvalue λ_i. The number of linearly independent eigenvectors corresponding to an eigenvalue is called the *geometric multiplicity*, g_i. If $a_i > g_i$, the eigenvalue is said to be *defective* and if $g_i > 1$, the corresponding eigenvalue λ_i is said to be *derogatory*. If a matrix has at least one defective eigenvalue, the matrix is said to be a *defective matrix*.

If an eigenvalue is defect, it does not have enough number of linearly independent eigenvectors to construct the matrix X. To get a complete base, so called *principal vectors* are used. For an eigenvalue λ_i with geometric multiplicity g_i, there exists g_i principal chains (one corresponding to each eigenvector),

$$(A - \lambda I)x_i^{(l)} = x_i^{(l-1)}, \text{ for } l = 2, 3, ..., h_i,$$

where h_i is the height (length) of the chain, l is the grade of a principal vector $x_i^{(l)}$, and $x_i^{(1)}$ is an eigenvector. For simple eigenvalues, $h = 1$ and $l = 1$. The relation above can also be written in matrix form as:

$$AX_i = X_i J_{h_i}(\lambda_i),$$

where $X_i = [x_i^{(1)}, x_i^{(2)}, ..., x_i^{(h_i)}]$ and $J_{h_i}(\lambda_i)$ is a Jordan block of size $h_i \times h_i$:

$$J_{h_i} = \begin{bmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_i \end{bmatrix}$$

For a general matrix A, with q distinct eigenvalues, $X = [X_1, X_2, ..., X_q]$ and q Jordan matrices $J(\lambda_i)$ can be formed. The matrix is then said to be in *Jordan normal form* (JNF),

$$X^{-1}AX = \text{diag}(J(\lambda_1), J(\lambda_2), \ldots, J(\lambda_q)),$$

where

$$J(\lambda_i) = \text{diag}(J_{h_1}(\lambda_i), J_{h_2}(\lambda_i), \ldots, J_{h_{g_i}}(\lambda_i)).$$

2.2 Matrix pencils and Kronecker structures

Given two matrices A and B of size $m \times n$, $A - \lambda B$ is called a *matrix pencil* . If $m = n$ and $B = I_n$ this corresponds to $A - \lambda I$. A generalized eigenvector and a generalized eigenvalue is a vector $x \neq 0$ and a scalar λ that satisfy

$$(A - \lambda B)x \equiv 0.$$

A general matrix pencil can have both left and right singular blocks as well as both finite and infinite eigenvalues (infinite if B is singular). The generalization of the JNF to matrix pencils is the *Kronecker Canonical Form* (KCF) . All $m \times n$ matrix pencils can be transformed into KCF [7]:

$$P^{-1}(A - \lambda B)Q = \text{diag}(L_{r_1}, \ldots, L_{r_p}, J(\lambda_1), \ldots, J(\lambda_k), L_{l_1}^T, \ldots, L_{l_q}^T),$$

where P ($m \times m$) and Q ($n \times n$) are nonsingular. $J(\mu_1), \ldots, J(\mu_k)$ form the *regular structure* and are Jordan blocks of the finite and infinite eigenvalues:

$$J_j(\lambda_i) \equiv \begin{bmatrix} \lambda_i - \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i - \lambda \end{bmatrix} \text{ and } J_j(\infty) \equiv \begin{bmatrix} 1 & -\lambda & & \\ & \ddots & \ddots & \\ & & \ddots & -\lambda \\ & & & 1 \end{bmatrix}$$

L_r and L_l^T correspond to the minimal indices of a singular pencil:

$$L_r \equiv \begin{bmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & -\lambda & 1 \end{bmatrix} \text{ and } L_l^T \equiv \begin{bmatrix} -\lambda & & & \\ 1 & \ddots & & \\ & \ddots & -\lambda & \\ & & 1 & \end{bmatrix}.$$

An $r \times (r+1)$ block L_r is called a *right singular block* (associated with a column minimal index) and an $(l+1) \times l$ L_l^T block is called a *left singular block* (associated with a row minimal index). L_r has the right singular vector $x_r^T = [1\ \lambda\ \lambda^2\ \ldots\ \lambda^r]$ such that $L_r x_r = 0$ for any $\lambda \in \mathbb{C}$. Similarly, L_l^T has a left singular vector $x_l = [1\ \lambda\ \lambda^2\ \ldots \lambda^l]$ such that $x_l L_l^T = 0$ for any scalar λ. The right and left singular blocks form the singular structure of $A - \lambda B$.

2.3 Matrix pairs

Matrix pairs (A,B) or (A,C), where $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$ and $C \in \mathbb{C}^{q\times n}$, appear in state-space systems, e.g., the controllability and observability pairs of matrices. For example, $[A\ B]$ (or $[A-\lambda I\ B]$) is the controllability pair (or pencil) of the system

$$\dot{x} = Ax + Bu,$$

where x is the state vector, A is the system matrix, u is the input vector and B is the input matrix of the system. Similarly,

$$\begin{bmatrix} A \\ C \end{bmatrix} \quad (\text{or} \begin{bmatrix} A-\lambda I \\ C \end{bmatrix})$$

is the observaibility pair (or pencil) of the system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

where y is the output vector and C is the output matrix of the system.

The stratification of the set of such pairs give qualitative (structural) information about the controllability and observaibility characteristics of nearby systems.

The canonical form consists of Jordan blocks and right singular blocks for (A,B), and Jordan blocks and left singular blocks for (A,C). An example when looking at the controllability pencil of a linear system is presented in Example 4 in Appendix A.

2.4 GUPTRI form

To compute the JNF or the KCF of a problem can be ill-conditioned. A denser canonical form, called generalized Schur-staircase form can however be computed using only unitary transformations. This form is a block upper triangular form that reveals the structure elements of the KCF or JNF.

The GUPTRI[1] form of a matrix pencil is a generalized Schur-staircase form that separates the singular and regular parts of the problem [2, 3]:

$$P(A-\lambda B)Q = \begin{bmatrix} A_r-\lambda B_r & * & * \\ & A_{reg}-\lambda B_{reg} & * \\ & & A_l-\lambda B_l \end{bmatrix},$$

where P ($m\times m$) and Q ($n\times n$) are unitary. The rectangular upper block triangular $A_r-\lambda B_r$ has only right minimal indices in its KCF, the same L_j blocks as $A-\lambda B$. Similarly, the $A_l-\lambda B_l$ only has left minimal indices in its KCF, the same L_j^T block as $A-\lambda B$. $A_{reg}-\lambda B_{reg}$ is an upper triangular block that contains all the finite and infinite eigenvalues of $A-\lambda B$.

Notice that the GUPTRI form can also be used for matrices and matrix pairs. For matrices this means that $B = I_n$, A is $n\times n$ and $P = Q^H$. Consequently, $A-\lambda B$ can have no left or right minimal indices or infinite eigenvalues and the GUPTRI form is therefore equal to $A_{reg}-\lambda B_{reg}$. Matrix pairs has the form $A = [F\ G]$ and $B = [E\ 0]$. When E is nonsingular, the KCF can have no left singular indices (L_j^T blocks).

[1] GUPTRI form is an acronym for Generilized UPer TRIangular form.

2.5 Orbits and bundles

Two square matrices, A and C, are said to be *similar* if there exists an invertible matrix P such that

$$C = P^{-1}AP.$$

The set of all matrices similar to a matrix A defines the *orbit* of the matrix, i.e.,

$$O(A) = \{P^{-1}AP | \det(P) \neq 0\}.$$

Notice that $O(A)$ consists of all matrices with the same eigenvalues and the same Jordan structure as A.

Two matrix pencils, $A_1 - \lambda B_1$ and $A_2 - \lambda B_2$, are said to be *strictly equivalent* if there exists non singular matrices, P and Q, such that

$$A_2 - \lambda B_2 = P^{-1}(A_1 - \lambda B_1)Q.$$

The set of all equivalent pencils to $A - \lambda B$ defines the *equivalence orbit* of the pencil, i.e.,

$$O(A - \lambda B) = \{P^{-1}(A - \lambda B)Q | \det(P)\det(Q) \neq 0\}.$$

$O(A - \lambda B)$ consists of all pencils with the same eigenvalues and the same KCF as $A - \lambda B$.

The equivalence orbits of the controllability and observability matrix pairs (pencils) are defined as follows:

$$\begin{aligned} O([F - \lambda I \; G]) &= \{[P^{-1}(F - \lambda I)Q \;\; P^{-1}G] \mid \det(P) \cdot \det(Q) \neq 0\} \\ O\left(\begin{bmatrix} F - \lambda I \\ H \end{bmatrix}\right) &= \left\{\begin{bmatrix} P^{-1}(F - \lambda I)Q \\ HQ \end{bmatrix} \mid \det(P) \cdot \det(Q) \neq 0\right\} \end{aligned}$$

A *bundle* is a union of orbits. If a matrix or pencil has the same canonical structure except that the distinct eigenvalues are different, they are said to be in the same bundle.

2.6 Codimension

The dimension of an orbit or bundle is equal to the dimension of its tangent space and is uniquely determined by the Jordan or Kronecker structure. To be specific, these orbits and bundles of matrices are manifolds in the n^2-dimensional space of $n \times n$ matrices. Similarly, the orbits and bundles of matrix pencils and matrix pairs are manifolds in spaces of dimension $2mn$ and $n(n+p)$, respectively. In practice, it is more convenient to work with the dimension of the space complementary to the tangent space, denoted *codimension*.

The difference between orbits and bundles are, that in a bundle, the eigenvalues are not specified, i.e., the tangent space of a bundle spans one extra dimension for each distinct eigenvalue compared to the corresponding orbit. In conclusion, the codimension of an orbit is the same as the corresponding codimension for the bundle plus the number of distinct eigenvalues.

EXAMPLE 1

For a generic $m \times n$ matrix pencil, $A - \mu B$, the tangent space of the bundle (and the orbit if $m \neq n$) spans the complete $2mn$ space. Therefore the codimension of that pencil is 0 ($\mathrm{cod}(A - \lambda B) = 0$). All degenerate pencils has a codimension greater that 0. The most degenerate matrix pencil (i.e., both A and B are all zeros) has a zero-dimensional tangent space and therefore the codimension of that pencil is $2mn$.

EXAMPLE 2

For a generic orbit of a $n \times n$ matrix, A, with n eigenvalues the codimension is n. Each specified eigenvalue limits the degree of freedom by one, i.e., the space complementary to the tangent space is of dimension n. It does not matter if A has multiple eigenvalues or not. If A gets rank deficient and hence less generic, the codimension further increases and for the zero matrix the orbit is zero-dimensional, i.e., the codimension is n^2.

2.7 Integer partitions

A partition, κ, of an integer n is a sequence $(k_1, k_2, k_3, ...)$ such that $k_1 + k_2 + k_3 + \ldots = n$ and $k_1 \geq k_2 \geq k_3 \geq \ldots \geq 0$. Further, an integer partition is said to dominate another partition λ, i.e., $\kappa > \lambda$ if $k_1 + k_2 + \ldots + k_i \geq l_1 + l_2 + \ldots + l_i$ for $i = 1, 2, \ldots$, where $\lambda \neq \kappa$. Different partitions of an integer can in this way form a dominance ordering. If $\kappa > \lambda$ and there is no partition μ such that $\kappa > \mu > \lambda$, then κ is said to *cover* λ.

2.8 Stratification

Edelman, Elmroth and Kågström [5] show how Jordan and Kronecker structures can be represented as integer partitions such that the closure relations of the various orbits and bundles are revealed by applying simple rules on these partitions. The closure relations or the closure hierarchy forms the *stratification* of Jordan and Kronecker structures. The stratification defines a partial ordering on orbits and bundles. One structure is said to cover another if its closure includes the closure of the other and there is no other structure in between.

A structure can never be covered by a less or equally generic structure. This implies that structures within the closure hierarchy can be ordered by their dimension (or their codimension). We remark that several orbits (or bundles) in a closure hierarchy can have the same codimension.

CHAPTER 3

Stratification by example

As mentioned before, the knowledge of stratification gives information about nearby structures that can be described as a connected graph[1].

3.1 A nilpotent matrix

To illustrate the theory of stratification we start by looking at a nilpotent matrix of size 6×6. Of course, any matrix with a single multiple eigenvalue μ can be made nilpotent by a shift, $A - \mu I$.

Consider a matrix $\tilde{A}$ with canonical structure $J_4(\mu) \oplus J_2(\mu)$, i.e., two Jordan blocks, one of size four and one of size two, both with the eigenvalue μ. Assume that the matrix is reduced to a GUPTRI form $A - \mu I = P^H(\tilde{A} - \mu I)P$:

$$A - \mu I = \begin{bmatrix} 0 & A_{12} & A_{13} & A_{14} \\ 0 & 0 & A_{23} & A_{24} \\ 0 & 0 & 0 & A_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix} = \left[\begin{array}{cc|cc|cc} 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.1}$$

Remark that the superdiagonal blocks A_{12}, A_{23}, and A_{34} of (3.1) have full column rank and define the "stairs" in the GUPTRI form. Each block A_{ii+1} is of size $m_i \times m_{i+1}$.

3.2 Canonical forms and Weyr characteristics

The m_i indices of the GUPTRI form reveal the Jordan normal form of the nilpotent matrix. In each step of the staircase algorithm, the nullspace of a deflated submatrix of $A - \mu I$ is computed. In the first step, $m_1 = \dim\mathcal{N}(A - \mu I)$ is computed and generally,

$$m_i = \dim\mathcal{N}(A - \mu I)^i - \dim\mathcal{N}(A - \mu I)^{i-1} \quad \text{for } i \geq 2.$$

[1] Thereby the name StratiGraph.

In other words, m_1 is the number of eigenvectors and m_i is the number of principal vectors of grade i associated with μ. In example (3.1), A has

$m_1 = 2$ A has 2 eigenvectors,
$m_2 = 2$ A has 2 principal vectors of grade 2,
$m_3 = 1$ A has 1 principal vector of grade 3,
$m_4 = 1$ A has 1 principal vector of grade 4,

and consequently

$$\begin{aligned} m_1 &= 2 = \dim\mathcal{N}(A-\mu I), \\ m_1+m_2 &= 4 = \dim\mathcal{N}(A-\mu I)^2, \\ m_1+m_2+m_3 &= 5 = \dim\mathcal{N}(A-\mu I)^3, \\ m_1+m_2+m_3+m_4 &= 6 = \dim\mathcal{N}(A-\mu I)^4. \end{aligned}$$

The number of principal vectors of grade i is equal to the number of Jordan blocks J_k of size $k \times k$ with $k \geq i$. The GUPTRI form (3.1), tells us that we have 2 Jordan blocks of size ≥ 1, 2 Jordan blocks of size ≥ 2, 1 Jordan block of size ≥ 3, and finally, 1 Jordan block of size ≥ 4, i.e., we have one Jordan block of size 2 and one of size 4.

The m_i indices are also referred to as the *Weyr characteristics*. For our example with canonical form $J_4(\mu) \oplus J_2(\mu)$, the Weyr characteristics associated with the eigenvalue μ are the following list of m_i values: $\mathcal{J}(\mu) = (2\ 2\ 1\ 1)$.

Without loss of generality, we assume that $\mu = 0$ in the following subsections.

3.3 Covered structures

We continue to investigate the GUPTRI form of our nilpotent example. Now, assume that A_{34} is almost rank-deficient. This means that α in (3.2) is a small number close to the machine precision. What happens if we instead consider this value to be zero?

$$\left[\begin{array}{cc|cc|cc} 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & \alpha \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{3.2}$$

In other words, how does the canonical structure change when $\alpha \to 0$? From (3.2), we see that m_3 will increase by 1 and m_4 will decrease by 1. The new GUPTRI form will be

$$\left[\begin{array}{cc|cc|cc} 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{3.3}$$

with the Weyr characteristics $\mathcal{J} = (2\ 2\ 2)$, indicating that there are no Jordan blocks of size 1 or 2, but two of size 3. The largest Jordan block has decreased in size and the second largest has grown. The matrix in (3.3) has the canonical structure $2J_3(0)$.

Next, let us consider the superdiagonal block A_{12} of the nilpotent matrix A in GUPTRI form (3.1). Since it has full column rank, A_{12} can be further reduced to an upper triangular (or even diagonal) form:

$$\left[\begin{array}{cc|cc|cc} 0 & 0 & \alpha & \times & \times & \times \\ 0 & 0 & 0 & \beta & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.4}$$

How does the canonical structure change if we consider A_{12} to be rank-deficient. For example, this happens if $\alpha \to 0$ in (3.4). Now, the canonical structure expressed in Weyr characteristics would be $\mathcal{J} = (3\ 1\ 1\ 1)$ with the GUPTRI form

$$\left[\begin{array}{ccc|c|c|c} 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.5}$$

The J_2 block has split into two 1×1 J_1 blocks. This matrix has the canonical structure $J_4(0) \oplus 2J_1(0)$.

Since we have imposed more rank deficiency in the original nilpotent matrix (3.1), the resulting GUPTRI forms (3.3) and (3.5) represent nilpotent matrices with more degenerate canonical forms. They also are said to be *less generic*.

In conclusion, we have made (3.1) less generic by changing different values into zero in (3.5) and (3.3). When looking at the space spanned by the orbit of the three matrices (i.e., the space spanned by the set of all similar matrices) they all span different "subspaces". However, the spaces spanned by the orbit of (3.5) and the orbit of (3.3) are both in the closure of the space spanned by the orbit of (3.1), i.e, $O(J_4(0)) \oplus J_2(0))$ *covers* both $O(J_4(0)) \oplus 2J(0))$ and $O(2J_3(0))$.

As mentioned before, any matrix with a single multiple eigenvalue μ can be made nilpotent by a shift, $A - \mu I$. This implies that the statement can be made even more general by saying that $O(J_4(\mu) \oplus J_2(\mu))$ covers both $O(2J_3(\mu))$ and $O(J_4(\mu) \oplus 2J_1(\mu))$.

3.4 Covering structures

We can also look at the two less generic nilpotent matrices the other way around. By introducing a small perturbation α in the GUPTRI forms (3.3) and (3.5), respectively, we obtain the original structure (3.1) in both cases.

What if we are interested in perturbations that make the original nilpotent matrix *more generic*? We can do this by adding a small but nonzero value α to (3.1) giving the following GUPTRI form:

$$\left[\begin{array}{cc|c|c|c|c} 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & \alpha & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.6}$$

The nilpotent matrix (3.6) has the Weyr characteristics is $\mathcal{J} = (2\ 1\ 1\ 1\ 1)$, i.e., the *covering* and more generic structure is $J_5(0) \oplus J_1(0)$. If we continue to add another small perturbation β so that all superdiagonal blocks are 1×1 and nonzero (i.e., of full rank), we get the following GUPTRI form

$$\left[\begin{array}{c|c|c|c|c|c} 0 & \beta & \times & \times & \times & \times \\ \hline 0 & 0 & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & \alpha & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.7}$$

The nilpotent matrix (3.7) corresponds to the most generic nilpotent matrix of size 6×6 with the canonical structure $J_6(0)$.

We have in this example by adding a small values to (3.1) made it more generic in (3.6) and (3.7). The orbit of (3.1) is in the closure of the orbit of (3.6) and that orbit is in the closure of the orbit of (3.7). From this follows that (3.1) also is in the closure of (3.7). We say that $\mathcal{O}(J_4(0)) \oplus J_2(0))$ is *covered by* $\mathcal{O}(J_5(0) \oplus J_1(0))$ and $\mathcal{O}(J_5(0) \oplus J_1(0))$ is *covered by* $J_6(0)$.

3.5 Canonical structure hierarchy

This example shows that with small perturbations, one can go from one canonical structure to another more generic structure. In this example, it is also clear that it is a very strict relation between them and by adding a small perturbation one can only go to a certain set of other structures. This relation forms a hierarchy of the canonical structures.

In Figure 2, the relationship between the discussed canonical structures using a graph representation, where the nodes represent orbits of the different matrices, is illustrated. In the discussion, we often only say matrix, structure, or we specify a canonical structure, when we formally mean the orbit (or bundle) of these objects.

The canonical structure of the original matrix, $A - \mu I$, is illustrated with double frames. Indeed, the graph shows the closure hierarchy between these five structures. We have shown that with a small perturbation one can go from $\mathcal{O}(2J_3(0))$ to $\mathcal{O}(J_4(0) \oplus J_2(0))$, i.e., $\mathcal{O}(2J_3(0))$ is *in the closure* of $\mathcal{O}(J_4(0) \oplus J_2(0))$ and $\mathcal{O}(J_4(0) \oplus J_2(0))$ *covers* $\mathcal{O}(2J_3(0))$. We also say that both $\mathcal{O}(2J_3(0))$ and $\mathcal{O}(J_4 \oplus 2J_1(0))$ are *covered by* $\mathcal{O}(J_4(0) \oplus J_2(0))$. With a small perturbation one can also go from $\mathcal{O}(J_4(0) \oplus J_2(0))$ to $\mathcal{O}(J_5(0) \oplus J_1(0))$. The closure relations above also imply, that with at small perturbation one can go from $\mathcal{O}(2J_3(0))$ to $\mathcal{O}(J_5(0) \oplus J_1(0))$ and further on to the most generic structure $\mathcal{O}(J_6(0))$.

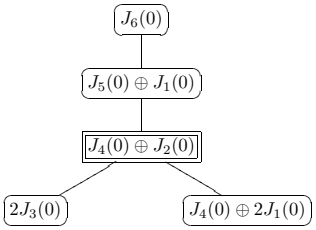

FIGURE 2: Relation between some canonical structures of a 6×6 nilpotent matrix orbit. The top three nodes in the graph correspond to the similarity orbits of (3.7), (3.6) and (3.1), respectively.

If there is a connected path in the graph from one node to another, then all structures on that path below a given node are in the closure of its orbit (or bundle). With a small perturbation it is always possible to go from a less generic structure to a more generic structure in the closure hierarchy. The complete closure hierarchy of the set of 6×6 nilpotent matrices is presented in Figure 3.

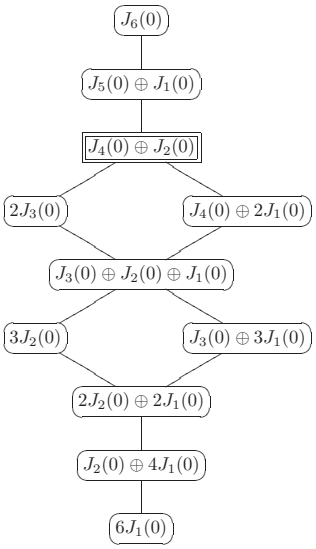

FIGURE 3: Jordan structure hierarchy of a 6×6 nilpotent matrix orbit.

CHAPTER 4

Describe your problem setup

When StratiGraph is started, a dialog window (Figure 4) is shown asking whether the user would like to open a blank window, an already saved graph or the graph wizard. The graph wizard helps to define your problem setup, including the starting structure. The input structure is the point of departure for investigating neighbouring structures above and below in the (closure) hierarchy.

FIGURE 4: The dialog window shown when StratiGraph is started.

4.1 Supported problem setups

The first step in the new graph wizard is to choose between the currently three supported problem setups, namely, matrices, matrix pencils, and matrix pairs (Both (A,B) and (A,C) cases). When a problem setup is selected, a brief explanation is shown (See Figure 5). Choose a current problem setup and continue to the next step of the wizard.

4.2 Fixed or nonfixed eigenvalues

As presented in Section 2.5, an orbit is for a matrix the set of all similar matrices, and for matrix pencils and matrix pairs, the set of all strictly equivalent matrix pencils or pairs.

A *bundle* is a union of orbits. If a matrix or pencil has the same canonical structure except that the distinct eigenvalues are different, they are said to be in the same bundle.

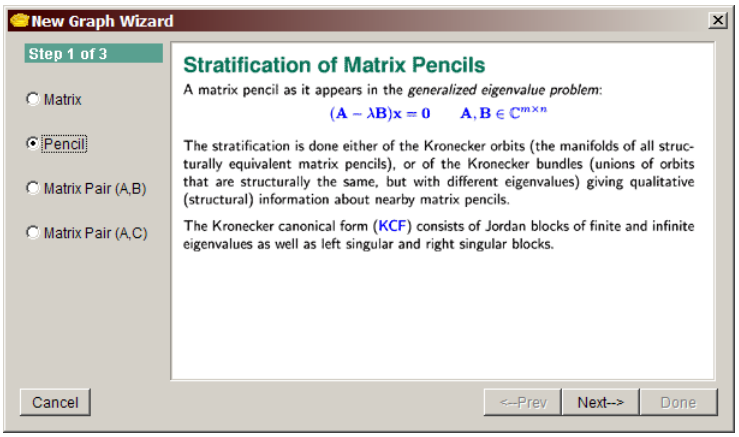
New Graph Wizard
Step 1 of 3
Matrix
Pencil
Matrix Pair (A,B)
Matrix Pair (A,C)
Stratification of Matrix Pencils
A matrix pencil as it appears in the generalized eigenvalue problem:
$(\mathbf{A} - \lambda\mathbf{B})\mathbf{x} = \mathbf{0} \qquad \mathbf{A}, \mathbf{B} \in \mathbb{C}^{m \times n}$
The stratification is done either of the Kronecker orbits (the manifolds of all structurally equivalent matrix pencils), or of the Kronecker bundles (unions of orbits that are structurally the same, but with different eigenvalues) giving qualitative (structural) information about nearby matrix pencils.
The Kronecker canonical form (KCF) consists of Jordan blocks of finite and infinite eigenvalues as well as left singular and right singular blocks.
Cancel
<--Prev
Next-->
Done

FIGURE 5: In the first step of the new graph wizard one can choose between different problem setups. Together with each setup a short description is shown.

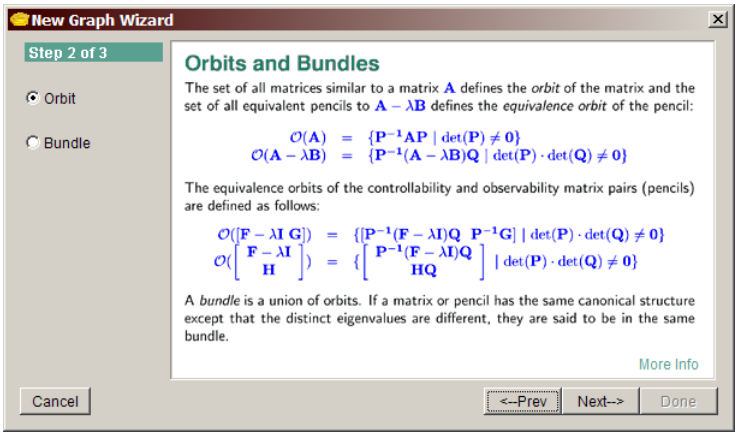
New Graph Wizard
Step 2 of 3
Orbit
Bundle
Orbits and Bundles
The set of all matrices similar to a matrix $\mathbf{A}$ defines the orbit of the matrix and the set of all equivalent pencils to $\mathbf{A} - \lambda\mathbf{B}$ defines the equivalence orbit of the pencil:
$\mathcal{O}(\mathbf{A}) = \{\mathbf{P}^{-1}\mathbf{A}\mathbf{P} \mid \det(\mathbf{P}) \neq 0\}$
$\mathcal{O}(\mathbf{A} - \lambda\mathbf{B}) = \{\mathbf{P}^{-1}(\mathbf{A} - \lambda\mathbf{B})\mathbf{Q} \mid \det(\mathbf{P}) \cdot \det(\mathbf{Q}) \neq 0\}$
The equivalence orbits of the controllability and observability matrix pairs (pencils) are defined as follows:
$\mathcal{O}([\mathbf{F} - \lambda\mathbf{I}\ \mathbf{G}]) = \{[\mathbf{P}^{-1}(\mathbf{F} - \lambda\mathbf{I})\mathbf{Q}\ \ \mathbf{P}^{-1}\mathbf{G}] \mid \det(\mathbf{P}) \cdot \det(\mathbf{Q}) \neq 0\}$
$\mathcal{O}\left(\begin{bmatrix} \mathbf{F} - \lambda\mathbf{I} \\ \mathbf{H} \end{bmatrix}\right) = \left\{\begin{bmatrix} \mathbf{P}^{-1}(\mathbf{F} - \lambda\mathbf{I})\mathbf{Q} \\ \mathbf{H}\mathbf{Q} \end{bmatrix} \mid \det(\mathbf{P}) \cdot \det(\mathbf{Q}) \neq 0\right\}$
A bundle is a union of orbits. If a matrix or pencil has the same canonical structure except that the distinct eigenvalues are different, they are said to be in the same bundle.
More Info
Cancel
<--Prev
Next-->
Done

FIGURE 6: In the second step of the new graph wizard one can choose between looking at the orbit or bundle stratification.

In practice, if the problem, for physical or other reasons, has a well-determined clustering of the eigenvalues then the orbit stratification is typically of interest. Otherwise, the bundle stratification is likely to be more useful. The choice between looking at the orbit or bundle stratification is done as illustrated in Figure 6, showing the second step in the wizard.

4.3 Problem size

In the third and final step of the new graph wizard, the size and starting structure is specified. One can choose between starting with the most or least generic structure of a given size, or a problem with a given canonical structure.

If the most or least generic structure is chosen, the size is specified directly and the corresponding canonical structure is automatically derived (See Figure 7).

If the canonical structure is known or at least presumed, this information can be entered by choosing "Other structure". Then the individual canonical blocks must be entered. In the matrix case, the Jordan blocks corresponding to each distinct eigenvalue are specified. For matrix pairs, the sizes of right singular blocks are also requested, and for matrix pencils also left singular blocks may appear. If no blocks of a specific type occurs in the canonical structure, the field is left empty. If the system has several blocks with the same size, they can also be entered as, e.g., `2*1`, meaning two blocks of size one.

In Figure 8, the canonical structure of a matrix pencil with one right singular block L_1 of size 1×2 and one Jordan block of size 1×1 is entered as the starting structure. The canonical structure, $L_1 \oplus J_1(\mu_1)$, is shown at the bottom left corner of the wizard window.

Notice that no explicit values of the distinct eigenvalues are requested or specified by the user.

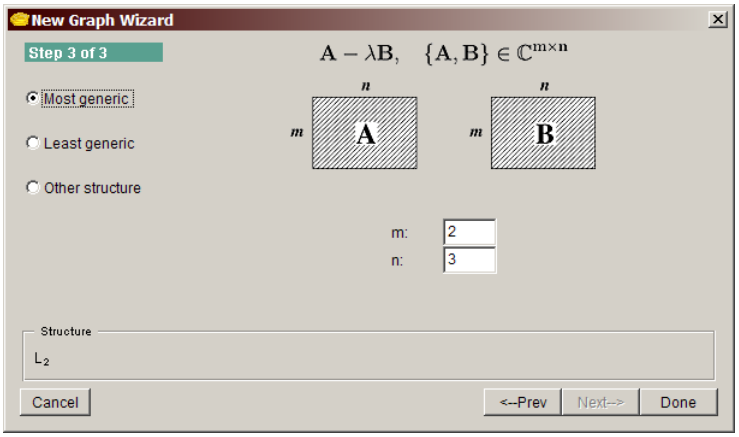

FIGURE 7: In the third step in the wizard, further information on the setup is given, such as the size and starting structure. Here the user has specified to look at the stratification of the orbit of a matrix pencil (Figures 5 and 6) of size 2×3, starting with the most generic pencil. The resulting canonical structure is shown at the bottom.

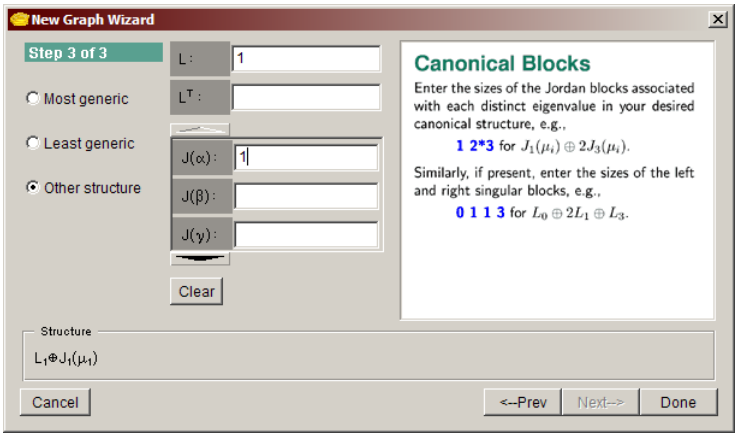

FIGURE 8: The canonical structure $L_1 \oplus J_1(\mu_1)$ is entered as the starting structure. Blocks are entered by their sizes and the resulting structure is shown at the bottom left corner of the wizard window.

CHAPTER 5

Expanding a graph

After the graph wizard is closed, the first node in the graph will appear in the main window. This node represents the orbit or bundle corresponding to the structure of the specified problem.

5.1 The node

On each node two numbers are displayed (Figure 9). The topmost number indicates the codimension of the corresponding orbit or bundle, i.e., the same number that also is visible in the left margin. The lowermost number indicates an order number that identifies individual nodes with the same codimension (and hence aligned on the same horizontal level in the graph).

By right-clicking on the node, all the information given about the structure (i.e., in most cases the canonical structure) will be presented. The information is shown until the mouse is clicked or the cursor is leaving the window. If the CTRL key is pressed, information about more then one node can be viewed at the same time.

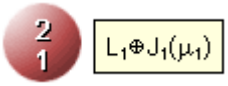

FIGURE 9: A node with the two numbers indicating the codimension and the order on that codimension level. This node indicates that the corresponding structure $L_1 \oplus J_1(\mu_1)$ has codimension 2 and that it is the first structure expanded with that codimension.

5.2 Expansion

The starting node is surrounded by a pattern looking like waves as can be seen in Figure 10. This node can now be expanded both upwards and downwards to see if and in that case

which other structures that are nearby. Small triangular indicators on the edge of the node signal whether the node can be expanded or not (Figure 10). A triangle at the top of the node indicates that there exists at least one unexpanded outgoing edge upwards. Similarly, if there is a triangle on the bottom edge of the node, then there exists at least one unexpanded outgoing edge downwards.

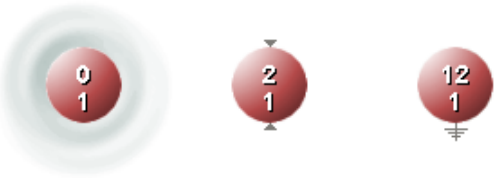

FIGURE 10: The three figures show different indicators a node can have. They are from the left; the starting node; a node that can be expanded both upwards and downwards; a leaf node that have no downwards going edges.

A node is activated and thereby coloured red by clicking on it. An active node can then be expanded upwards by clicking on the upper half of the node, and downwards by clicking on the lower half of the node. Any node connected to an active node is coloured blue and the rest of the nodes are green.

Nodes in the graph are ordered vertically by their codimension. If two nodes represent different structures but with the same codimension they are aligned on the same level. In the left margin of the graph window the codimension for each displayed level of nodes are shown (See Figure 11).

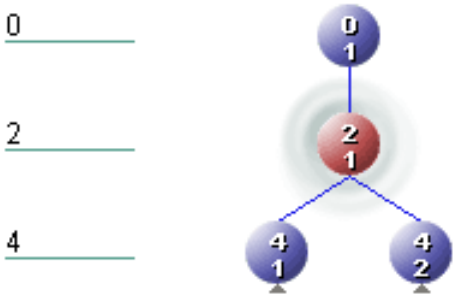

FIGURE 11: Here the starting node has been expanded one step both upwards and downwards. Notice how the nodes are aligned vertically and horizontally by their codimension.

If there is no triangle at the edge of a node, no further outgoing edges exists and nothing will happen if one try to expand it. A node with no downwards going edges is called a leaf node and they are indicated by a stylized arrow head (see Figure 10).

The graph can also be expanded recursively, i.e., when a node is expanded downwards (or upwards) it will continue to expand the covered (or covering) nodes downwards (or upwards). This will proceed until no new nodes can be found. Also the complete graph can be expanded. During this process a dialog window will appear (Figure 12) showing the progress of the expansion and it also provide a way to halt the expansion process. Notice that even a fairly small initial structure can produce a huge graph and consequently will take

a lot of memory and computer resources to compute. Recursive expansion is therefore only recommended on small graphs.

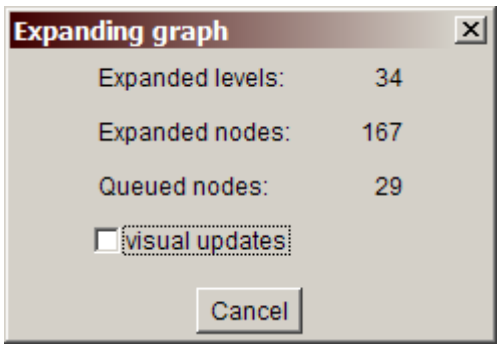

FIGURE 12: Dialog window shown when recursive expansion is done. Already for moderate problem sizes this could be a time and memory consuming operation.

Recursive expansion can be found using the menu under the item “Graph” and also on the button bar as the buttons depicting several levels of nodes. A complete description of all the menu choices and menu buttons is presented in Appendix B.

5.3 The relation between neighbouring nodes

An edge between two nodes indicates that the nodes’ corresponding structures are forming a covering relation in the space of matrices (matrix pencils, or matrix pairs).

The edges themselves show the canonical structure transitions between two connected nodes. When right clicking on an edge, information on which blocks in the upper connected node that have changed in order to get the lower connected node is presented. Notice that only blocks that have changed are presented, not the complete canonical structure. An example is displayed in Figure 13. We refer to Edelman-Elmroth-Kågström [5] for the rules of canonical structure transitions in the stratification process.

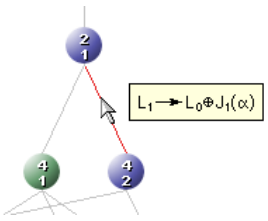

FIGURE 13: Here $L_1 \oplus J_1(\mu_1)$ covers $L_0 \oplus J_1(\mu_1) \oplus J_1(\mu_2)$. By right-clicking on the edge, we can see that the L_1 block has changed to $L_0 \oplus J_1(\mu_2)$.

5.4 Rearrange a graph

The algorithm that places the nodes tries to get as few edge-crossings as possible. However, sometimes it is desirable to change the placements of the nodes due to estethical reasons or

because a group of nodes have a special relation. Nodes can not change its vertical positions (they can not change codimension), but they can be moved horizontally.

A node is moved by dragging them, holding the left mouse button. When a node is dragged, a node without identification numbers will appear. Moving the mouse sideways will move the node. The node is given the new position when the button is released. The node will snap into some positions at a regular distance to help rearranging the graph nicely. The node numbers are left unchanged by this operation.

5.5 Structure overview in separate windows

For better overview of how structures are related, a separate window can be opened that only displays a subset of the generated structures, i.e., the current active node and the closest related nodes (Figure 14). Nodes with a lower codimension connected to the active node (i.e., the active node is in their closures) have a blue arrow pointing upwards in front of them. Structures with a higher codimension (i.e., in the closure of the active node) and connected with the active node have a blue arrow pointing downwards. The currently active node has a red arrow pointing to the right in front of it. Above each node both their codimension and order in their codimension layer are displayed, i.e., the same information as displayed on the graph node itself.

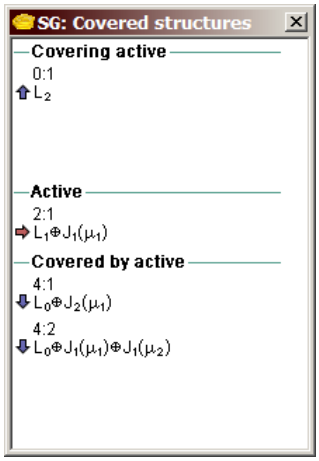

FIGURE 14: Window showing covering, active and covered structures.

Another window displays a list with all the generated nodes (Figure 15). The same set of symbols as in the window above are displayed in front of the nodes. For nodes not directly connected to the currently active, a green bullet is displayed in front of the structure.

To activate one of the structures and make the corresponding node active in the main window, click on it with the mouse pointer.

5.6 Scaling and overview of the graph

The total number of nodes grows exponentially with the size of the problem. Already for moderate-sized problems the total number of structures and thereby the number of nodes

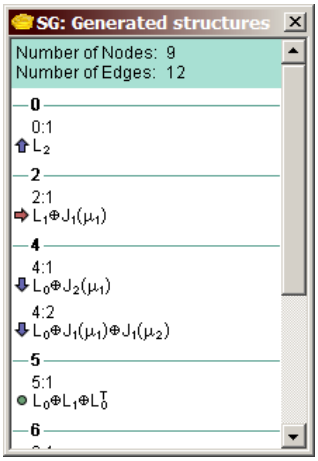

FIGURE 15: Window showing all generated structures.

in the graph can be very large. In these cases, it is therefore not often useful to look at the complete graph but more likely a subset of the graph. But even in those cases the graph window can be too small to fit all the desired nodes. In those cases, the built in scaling functionality can be of use. The graph can be scaled down to 1% and up to 200% of the original size.

The scaling tool can be found in the tool bar with the button depicting a magnifying glass or in the menu under ”View → Zoom”. The dialog window opened (Figure 16) looks like an arrow where the desired scale factor is adjusted by dragging the mouse pointer up or down. When the mouse button is released the graph will be rescaled.

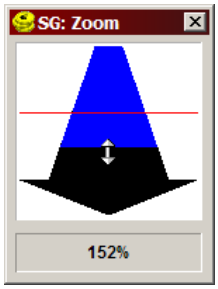

FIGURE 16: The scale is set by pressing the left mouse button and adjusting the level from 1 to 200 %.

Another way to better adjust the scaling is to automatically rescale the graph so that it fits the current graph window. This is done by clicking on the button in the tool bar depicting a magnifying glass surrounded by three small arrows or in the menu under ”View → Zoom to fit”. Then the graph is either zoomed down or up to best fit the graph window.

CHAPTER 6

Additional functionality

6.1 Exporting and printing a graph

StratiGraph can export graphs and structures to PostScript-files that can be viewed or printed. A print dialog window (Figure 17) is opened by clicking on the button depicting a printer in the tool bar or in the menu under ”File → Print”. In the dialog window one can specify which information to export, i.e., either only the graph, only the structure information or both. Also the paper size its orientation (landscape or portrait format) can be specified.

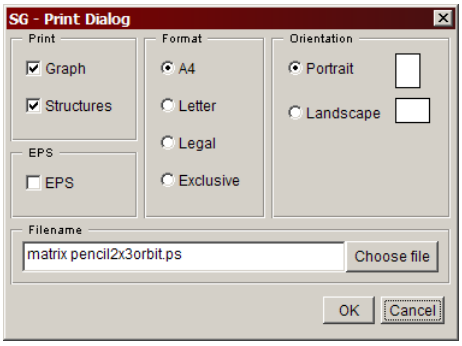

FIGURE 17: The dialog window for exporting a graph with corresponding structure information to PostScript.

The graph can also be exported as an Encapsulated PostScript file (EPS). This is done by selecting the EPS check-box. In this case the node information can not be included, only the graph.

Also, the name of the file to which the graph will be exported must be given. A default name will be suggested and then the file will be saved in the current working directory, but both the name and location may be changed either manually or with a file-dialog window.

6.2 Saving and loading a graph

When working on a problem setup, one may want to save an expanded graph for later use. By clicking on the button depicting a floppy disk or using the menu under "File → Save", a file dialog window is opened where a name for the saved graph can be specified.

A graph is loaded by clicking on the button depicting an opening folder or using the menu under "File → Open". Then previously saved files can be browsed. Under the "File" menu a list of previously saved files are also presented. By clicking on one of these, the graph is automatically loaded.

6.3 Different notation and font sizes

The default notation for canonical structure information is *block structure notation* where the individual blocks are presented like $L_0 \oplus J_1(\mu_1)$, representing a right singular block of size zero (0×1) and one Jordan block of size one (1×1) with the associated eigenvalue μ_1.

Two other notations are currently also available, namely Weyr characteristics and Segre characteristics. The Weyr characteristics are discussed in Section 3.2. In Example 3 and Example 4 the two characteristics are illustrated. Figure 18 shows how all three notations are used in StratiGraph. The notation currently used is shown in the status bar of the main window.

EXAMPLE 3

The structure

$$L_1^T \oplus L_0^T \oplus J_3(\mu_1) \oplus 2J_1(\mu_1)$$

is written as

$$\mathcal{L} = (2\ 1) \qquad \mathcal{J}(\mu_1) = (3\ 1\ 1)$$

in Weyr characteristics. $\mathcal{L} = (2\ 1)$ says that there are two left singular bocks L_k^T with $k \geq 0$ and one L_k^T with $k \geq 1$, i.e., there is one left singular block L_0^T and one block L_1^T. $\mathcal{J}(\mu_1) = (3\ 1\ 1)$ represents three Jordan blocks J_k with $k \geq 1$, one block with $k \geq 2$ and finally one block with $k \geq 3$ all corresponding to the same eigenvalue μ_1. This means that we have two $J_1(\mu_1)$ blocks and one $J_3(\mu_1)$ block.

Segre characteristics list the sizes of the canonical blocks in decreasing order.

EXAMPLE 4

The structure

$$L_1^T \oplus L_0^T \oplus J_3(\mu_1) \oplus 2J_1(\mu_1)$$

is written as

$$\mathcal{L} = (1\ 0) \qquad \mathcal{J}(\mu_1) = (3\ 1\ 1)$$

in Segre characteristics. $\mathcal{L} = (1\ 0)$ says that there are one left singular bock L_k^T of size 1 (1×2) and one of size 0 (0×1). $\mathcal{J}(\mu_1) = (3\ 1\ 1)$ says that we have one Jordan block of size 3 and two Jordan blocks of size 1 all corresponding to the same eigenvalue μ_1.

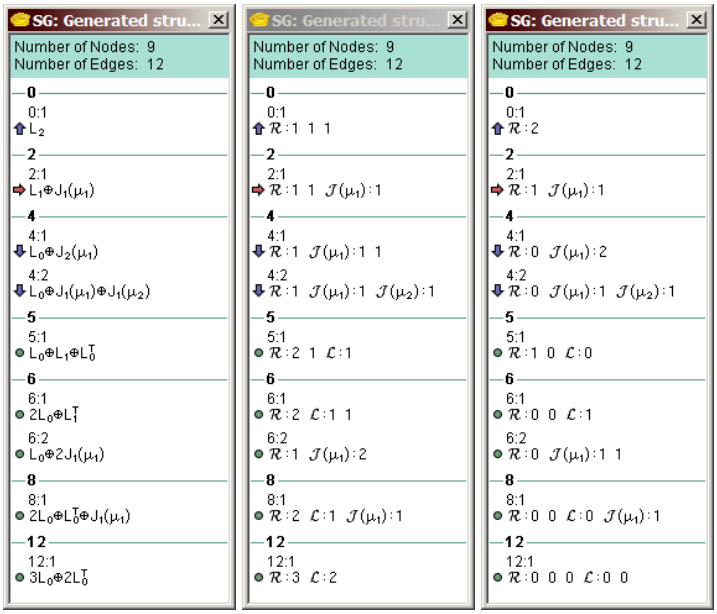

FIGURE 18: The same structure information given with different notations. From left to right the structures are shown in block structure notation, Weyr characteristics, and Segre characteristics.

Besides choosing between different notations of the structures, one can change the displayed font size. Three different sizes are available, namely 12pt, 14pt and 16pt. The font sizes are available under the menu ”Options → Font size”.

6.4 Viewing options

Besides the ones already described, a number of options for the graph are available.

The horizontal as well as the vertical distance between nodes can be changed. This can make a very broad graph more compact or a graph with lots of connections more easy to overview. The distance is changed under ”Options → Node distance”. A small window is then opened where the values can be changed (Figure 19).

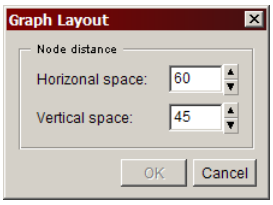

FIGURE 19: The distance, both horizontal and vertical, between the nodes can be adjusted to make the graph more or less compact.

The symbols describing non-expanded nodes, leafs and starting node can be turned on and off. This is done under ”View → Decorations”.

A graph can also be displayed in a more compact way. In a compact layout all extra

space between two codimension layers where there are no nodes are not visualized. The compact layout can be selected under ”View → Compact graph layout”.

6.5 Loading Plug-ins

Two different kinds of plug-ins can be loaded into StratiGraph, i.e., either a *program extension* or a new *problem setup*. The extensions add some kind of functionality to the software, e.g., the ability to communicate with third party software. Currently, there are three different kinds of problem setups supported by StartiGraph, four if you count the (A,B) and (A,C) matrix pair setups as different. As the research in this area continues it is likely that more setups will be added like matrix triples (A,B,C) etc. StratiGraph already supports the addition of new setups that can be loaded as a plug-in with the plug-in manager. The Plug-in manager is opened under ”Options → Plugin manager” and the window is shown in Figure 20.

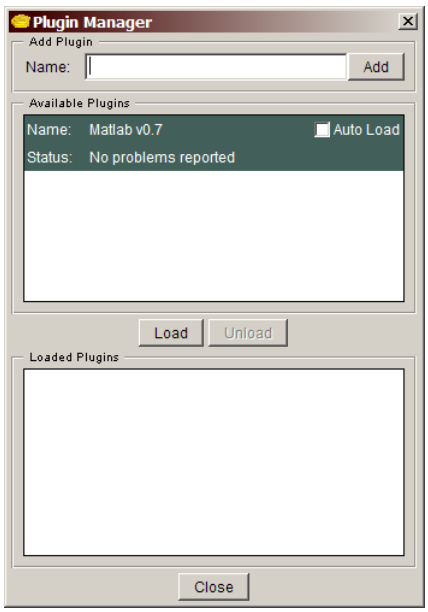

FIGURE 20: The plug-in manager window. A plug-in called “Matlab” is identified by the program and is ready to be loaded.

If you like to use a new plug-in, first follow the installation instructions with the plug-in. In the top text field, enter the name of the plug-in and press the “Add” button. If the plug-in is found an information text will appear under “Available Plug-ins” and the plug-in is now ready to be loaded. Mark the plug-in by clicking on the information text and then press the “Load” button. The plug-in is then loaded into the program and can be used. The plug-in information will also be moved to the “Loaded plug-ins” list. To unload a plug-in, select the information text and press the “Unload” button.

Next time StratiGraph is started the plug-in will be automatically identified and added to the “Available Plug-ins” list, but not loaded. If you like that plug-in to also be loaded, there is a check-box on the right that can be marked.

CHAPTER 7

Limitations and future developments

7.1 Current limitations

When expanding even small problems the total number of nodes in the graph can be very large. The number of nodes grows exponentially with the problem size. A large graph requires more memory and computer resources to construct. A very large graph also does not provide much useful information (though it is interesting to view). Therefore one might need to be careful in the expansion process especially when it comes to recursive expansion. Try to only expand parts of the graph that are of interest.

StratiGraph is run with a Java virtual machine (JVM), and usually a JVM restricts how much memory that can be used by the software. If memory problems occur, refer to the JVM manual for that computer platform.

The only way to export the expanded graph is to Post Script (PS) or Encapsulated PS (EPS). However, these formats can be converted to PDF or different image formats using appropriate software.

7.2 Supported problem setups

Ongoing research with new knowledge in this field is constantly added into StratiGraph. Currently StartiGraph supports matrices, matrix pencils and matrix pairs. The ambition is to add functionality for matrix triples and matrix quadruples.

7.3 Expansion options

When the graphs get very large it would be useful to collapse or even remove parts of the graph to be able to focus on different aspects of the problem. Also the ability to expand the graph to find specific structures or highlight specially interesting subsets of nodes is also planned to be added.

7.4 Quantitative results

StratiGraph in this version focuses on qualitative results, i.e., the stratification process and associated structures. One major field of interested connected to the stratification is different quantitative information given specific input data. Plug-ins that communicate with the Matlab environment are developed and are planed to be incorporated in a future version of StratiGraph. Given a set of data in Matlab, StratiGraph then provides the stratification graph together with quantitative information, including distances to nearby structures.

APPENDIX A

Examples

In the following examples, we illustrate the complete stratification of some small-sized problems including different problem setups.

Example 1 – Nilpotent matrix

In Section 3, we considered a nilpotent matrix when introducing stratification by example. Now we illustrate the same example using StratiGraph.

A nilpotent matrix has only one eigenvalue (that is 0). Also, this eigenvalue can never be changed, because, then the matrix would not be nilpotent anymore. Therefore, when looking at nilpotent matrices only the orbit case (fixed eigenvalues) are of interest and only matrices with one eigenvalue. Matrices with only one eigenvalue can be made nilpotent by shifting the matrix with its eigenvalue ($A - \mu I$).

Looking at the same example as in Section 3 in StratiGraph, we specify a matrix-case, and choose "Other structure" and "Orbit". We specify the Jordan structure $(4\ 2)$ for one eigenvalue (Figure 21).

When finishing the wizard, the starting node will appear in the middle of the graph window. Following the example in Section 3, the graph is first expanded downwards by clicking on the lower part of the node. This will create the two closest downward neighbours, the same structures as in the staircase forms (3.3) and (3.5), namely $2J_3(\mu_1)$ and $J_4(\mu_1) \oplus 2J_1(\mu_1)$. Similarly, expanding the starting node upwards gives the same structure as in (3.6), namely $J_5(\mu_1) \oplus J_1(\mu_1)$.

After the node expansion is repeated and completed, the same graph as in Figure 3 appears (Figure 22).

Example 2 – Matrix as orbit and bundle

In the next example, we look more closely on the difference between an orbit and a bundle case. When looking at structures in a given orbit, the eigenvalues do not change. If the canonical structure contains singular blocks, eigenvalues can disappear and appear, but one

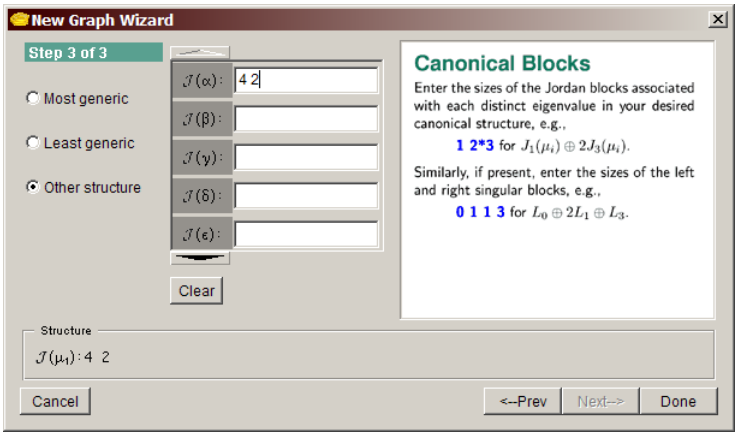

FIGURE 21: For a nilpotent matrix we specify the matrix case and looks at the stratification of the orbit. The block sizes are given for a nilpotent matrix with the canonical structure $J_4(0) \oplus J_2(0)$. Notice that we only have one (non-specified) eigenvalue.

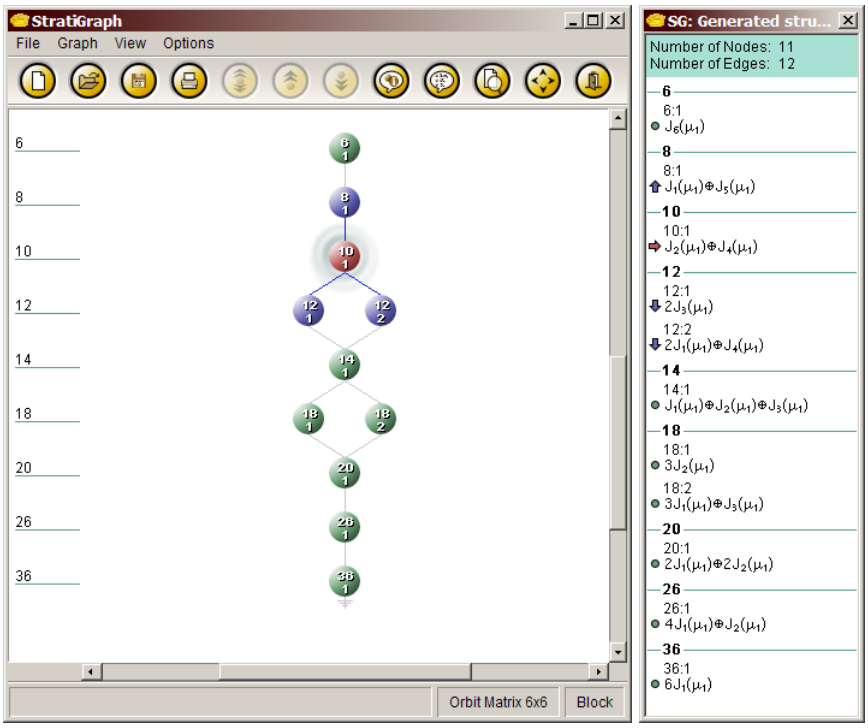

FIGURE 22: The orbit stratification of a 6×6 nilpotent matrix. In the left window, the graph is expanded upwards and downwards starting from $J_4(0) \oplus J_2(0)$. In the right window, all generated structures are displayed.

existing eigenvalue can not change into another or split into different ones. When looking at bundles, the eigenvalues can however change and coalesce, and we can get new ones or loose some.

The example is a 4×4 matrix. The initial canonical structure has two Jordan blocks of size 2 and 1 corresponding to one eigenvalue μ_1, and one Jordan block of size 1 corresponding to μ_2.

$$J_2(\mu_1) \oplus J_1(\mu_1) \oplus J_1(\mu_2).$$

We first consider the matrix orbit case, where all eigenvalues are kept fixed. The single Jordan block corresponding to μ_2 can obviously not be changed, but the regular 3×3 part corresponding to μ_1 can change. The 2×2 Jordan block $J_2(\mu_1)$ can split into $J_1(\mu_1) \oplus J_1(\mu_1)$ (less generic) or $J_1(\mu_1) \oplus J_2(\mu_1)$ can merge into a 3×3 Jordan block $J_3(\mu_1)$ (more generic). The complete orbit stratification is shown in Figure 23.

In a bundle case however, we get many more possibilities. Of course, the same cases that appear in the orbit case exist also in the bundle case. The eigenvalues can change but does not have to. For example, we get a more generic case when $J_2(\mu_1)$ splits into two Jordan blocks with one new eigenvalue creating a matrix with the canonical structure $2J_1(\mu_1) \oplus J_1(\mu_2) \oplus J_1(\mu_3)$. The most generic case is a matrix with four Jordan blocks, all with different eigenvalues, i.e., the typical case for a random matrix.

Considering more degenerate neighbours, we also get different new structures that do not appear in the orbit case. For example, we can have two eigenvalues coalescing into the same eigenvalue and we get a $J_3(\mu_1)$ block.

The complete graph for this small bundle example is shown in Figure 24. Already in fairly small examples the total number of different structures can be plenty. However, usually one is not interested in the complete graph, but small subgraphs which show the structures in the vincinty of the starting node.

Example 3 – Matrix pencils

In this exampel, we look at a matrix pencil of size 3×5, i.e.,

$$A - \lambda B$$

where $A, B \in \mathbb{C}^{3\times 5}$. We are starting with the most generic structure that is $L_1 \oplus L_2$ in both the orbit and bundle cases, i.e.,

$$A - \lambda B = \left[\begin{array}{cc} L_1 & \\ & L_2 \end{array}\right] = \left[\begin{array}{cc|ccc} -\lambda & 1 & 0 & 0 & 0 \\ 0 & 0 & -\lambda & 1 & 0 \\ 0 & 0 & 0 & -\lambda & 1 \end{array}\right] \tag{A.1}$$

The least generic pencil is of course when both A and B are zero matrices, that is, $A - \lambda B = 5L_0 \oplus 3L_0$. In both the orbit and bundle cases, we get the same set of nodes in this example but the way they are related differs.

Eigenvalues can coalesce and split in the bundle case, but not in the orbit case. However in the orbit case we can see new eigenvalues emerge by structure transitions in the singular blocks [5].

The largest regular part in this 3×5 example is 3×3. In an orbit case, the most interesting thing is if this regular part only has one eigenvalue. In that case the most generic structure is $J_3(\mu)$ that covers $J_2(\mu) \oplus J_1(\mu)$ that in turn covers the least generic structure $3J_1(\mu)$.

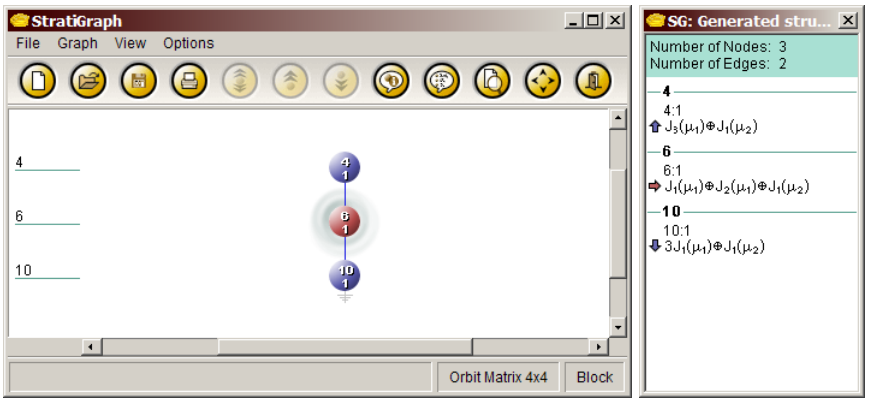

FIGURE 23: The orbit stratification of a 4×4 matrix.

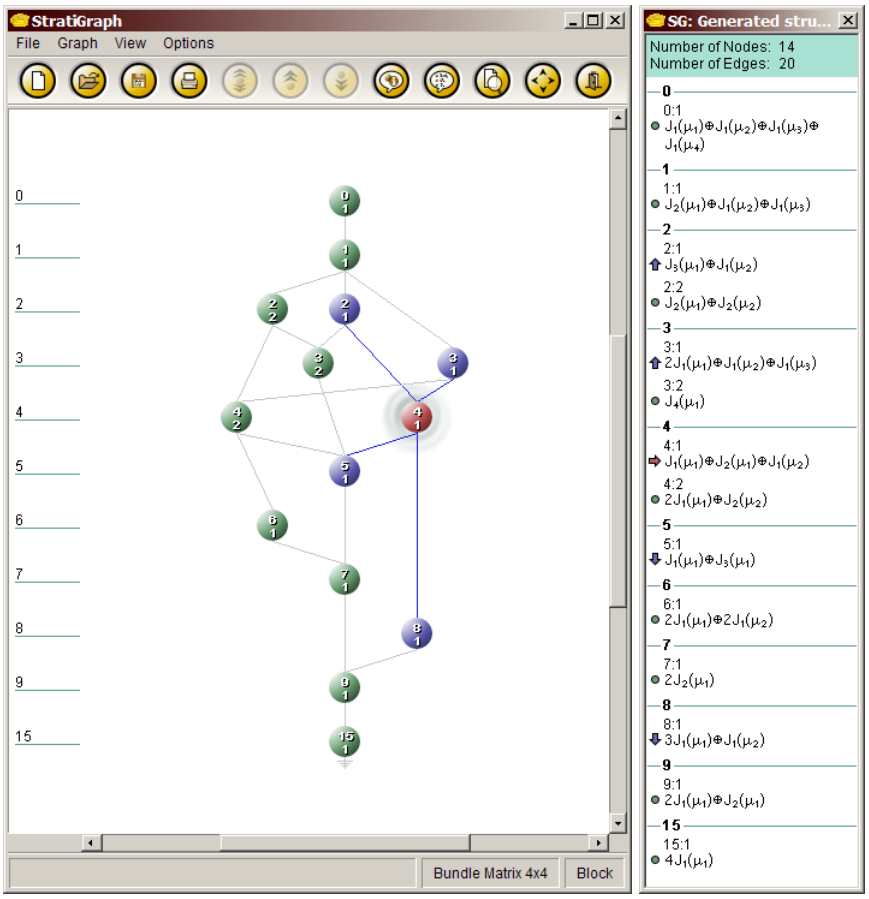

FIGURE 24: The bundle stratification of a 4×4 matrix.

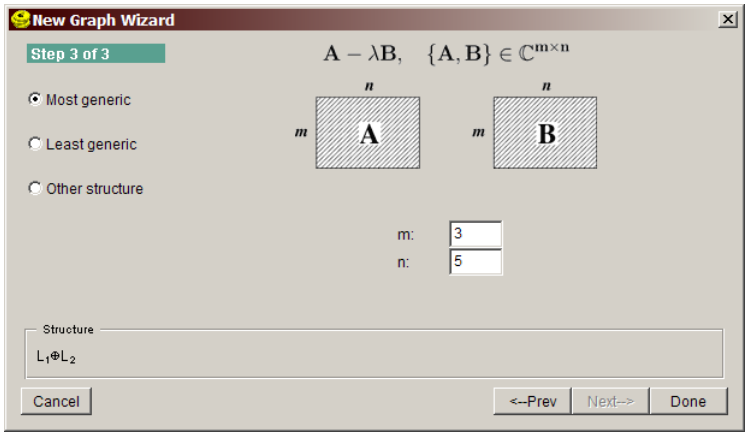

FIGURE 25: To view the stratification of a 3×5 matrix pencil we start with the most generic pencil $L_1 \oplus L_2$.

This small connected graph can also be found as a subgraph in the matrix pencil as seen in Figure 26. If the regular part instead has two distinct eigenvalues, this also forms a small subgraph with $J_2(\mu_1) \oplus J_1(\mu_2)$ that covers $2J_1(\mu_1) \oplus J_1(\mu_2)$. However, since the eigenvalues can not change these two subgraphs are never connected. That is, with a small perturbation one can not go from one of the subgraphs to the other and still be in an orbit with a covering relation. The same apply to the third subgraph (with only $J_1(\mu_1) \oplus J_1(\mu_2) \oplus J_1(\mu_3)$) that corresponds to the case when the regular part has three distinct eigenvalues.

Then looking at the bundles of the same pencil, the regular part behaves differently. Since the eigenvalues can coalesce and split, the whole 3×3 regular part is connected in one subgraph (See the right graph in Figure 26). That is, when looking at bundles, one can with a small perturbation go from a canonical form with multiple eigenvalues to a canonical form where they have split into distinct eigenvalues and still be in a bundle with a covering relation with the original.

Example 4 – Control application

Several characteristics of linear state space models , such as controllability and observability, can be described in terms of matrix pencils (e.g., see [11, 1, 3]). Consider the linear system

$$E\dot{x}(t) = Ax(t) + Bu(t),$$

where E and A are $n \times n$ and B is $n \times p$. The system is *controllable* if, starting with $x(0) = x_0$, it is possible to choose the input u to bring the state vector$x$ to an arbitrary state in some finite time t_N. One way to characterize controllability is via the *controllability pencil*

$$\mathcal{C}(E, A, B) = [B|A] - \lambda[0|E].$$

It has full rank except at $k < n$ values of λ, which correspond to the *uncontrollable modes* of the linear system above.

Controllability decision is an ill-posed problem in the sense that small perturbations in the data may drastically change the behavior of the system. Therefore, it is of great value to

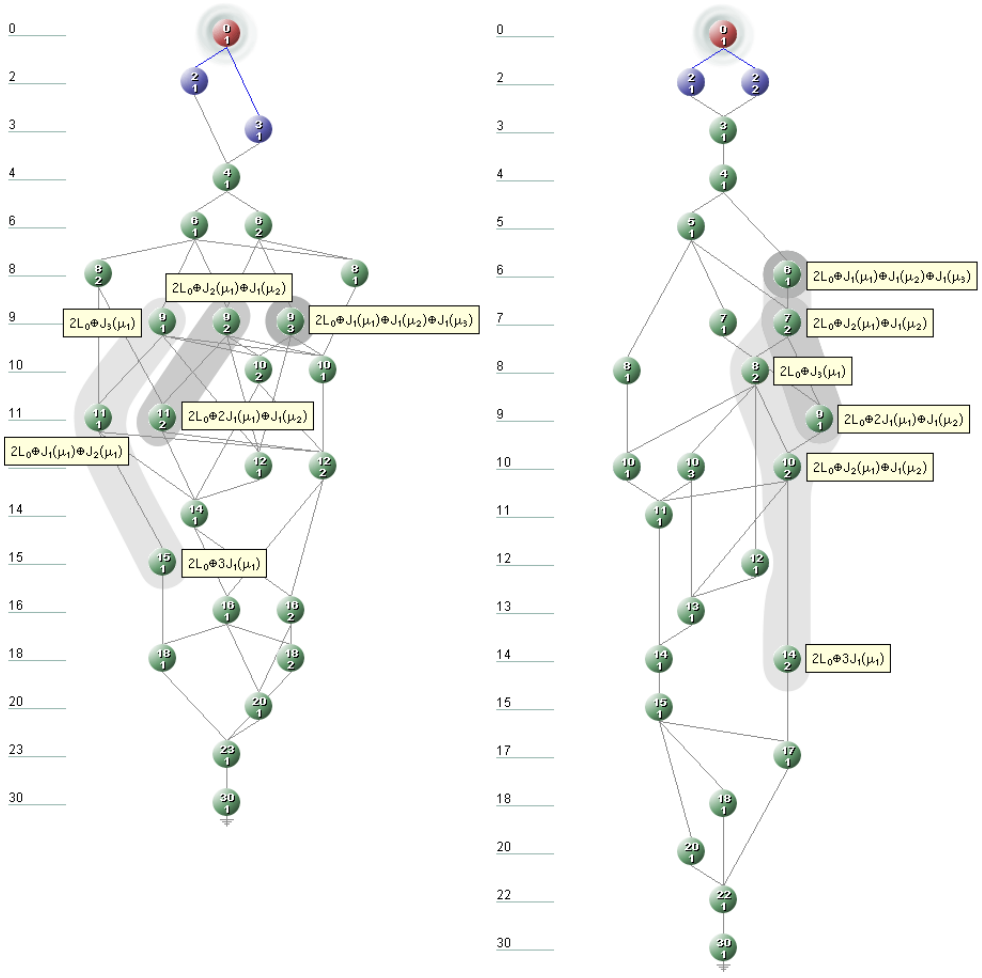

FIGURE 26: The complete graph illustrating the stratification of a 3×5 matrix pencil. On the left the orbit case is shown and on the right the bundle case is shown. In both graphs, the subgraphs of the pencils with a 3×3 regular part are highlighted. The other nodes have no regular part or a regular part of size 1×1 or 2×2. The figures illustrate how eigenvalues can change and merge in the bundle case but not in the orbit case.

know the stratification of the controllability pencil and thereby get qualitative information of nearby canonical structures.

Consider a system with three states ($n = 3$) and two inputs ($p = 2$), i.e., a 3×5 controllability pencil. In the left part of Figure 27, the graph for the bundle stratification of the set of matrix pencils of size 3×5 is displayed. This gives the complete picture, but several of the Kronecker structures represented in the graph are not possible for this application. Let $\mathbf{A} \equiv [B|A]$ and $\mathbf{B} \equiv [0|E]$, and we assume that E is nonsingular. Then it follows that $\mathbf{A} - \lambda\mathbf{B}$ can only have L_j blocks and finite eigenvalues. The substratification corresponding to these cases are displayed in the top-right part of Figure 27, starting from the node labeled "0 over 1" and ending at the node labeled "14 over 2". The generic case is $L_1 \oplus L_2$, which corresponds to a controllable system. This graph is obtained by treating the controllability pencil as a matrix pair, which in this case automatically rules out some cases that appear in the stratification of the set of 3×5 matrix pencils.

Notice that the codimension of the bundles in both graphs are the same but not their dimensions. The dimension of the bundle of the most generic pencil $L_1 \oplus L_2$ (codimension 0) has the dimension 30 ($2mn$) in the matrix pencil stratification, but dimension 15 ($n(n+p)$) in the matrix pair stratification (Section 2.6).

We remark, that at codimension level 2 there are two structures $L_0 \oplus L_3$ and $2L_1 \oplus J_1(\mu_1)$. The last case represents a system with one uncontrollable mode, and a controllable subspace of size two. The structure $L_0 \oplus L_3$ represents a system which is controllable, but only for one of the two input variables.

The least generic case $2L_0 \oplus 3J_1(\mu_1)$ corresponds to a system with three multiple uncontrollable modes.

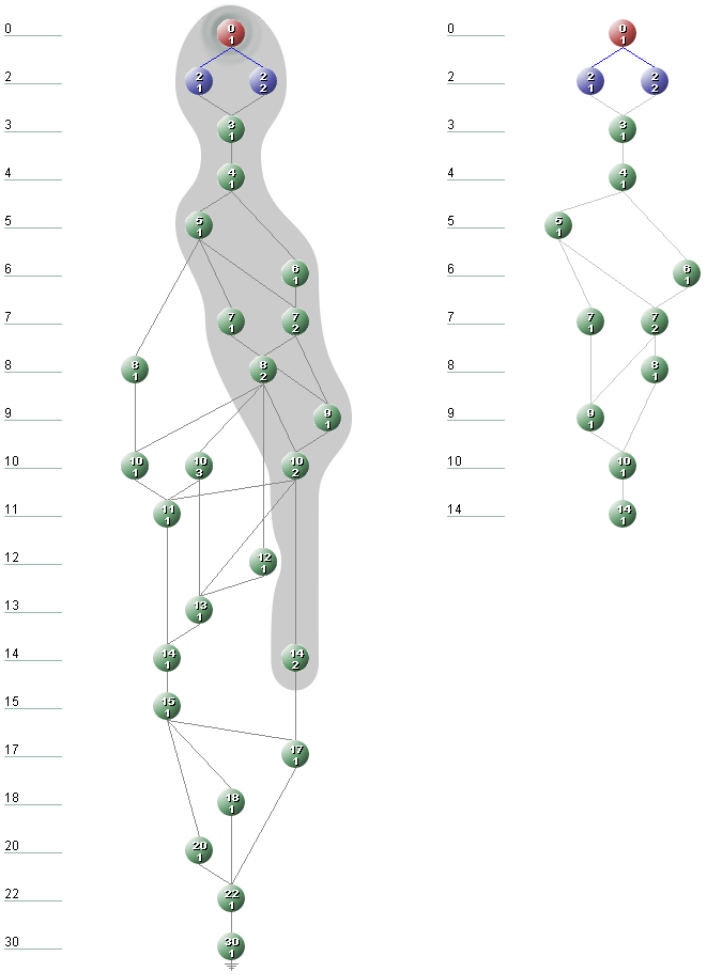

FIGURE 27: To the left, the complete graph illustrating the bundle stratification of a 3×5 matrix pencil is shown. The highlighted subgraph contains the nodes corresponding to the possible canonical forms when looking at the pencil as a matrix pair. That subgraph is identical to the graph illustrating the bundle stratification of a 3×5 matrix pair that is show on the right.

APPENDIX B

StratiGraph commands

Menu bar

▷ **File**

- ▷ **Open** Open a saved graph.
- ▷ **Save** Save the graph to file.
- ▷ **Print** Export the graph to PostScript.
- ▷ **About** Shows version and author information.
- ▷ **Exit** Exits StratiGraph.

▷ **Graph**

- ▷ **New graph**
  Open the graph wizard.
- ▷ **Set active as start node**
  Set the active node as starting node. Has meaning in some plug-ins.
- ▷ **Expand upwards**
  Expand the active node upwards.
- ▷ **Expand recursive upwards**
  Expand the active node upwards recursively.
- ▷ **Expand downwards**
  Expand the active node downwards.
- ▷ **Expand recursive downwards**
  Expand the active node downwards recursively.
- ▷ **Expand complete graph**
  Expand the complete graph.

▷ **View**

- ▷ **Show covers**
  Open a window that shows the closest neighbouring nodes to the active node. Off as default.

- ▷ **Show generated structures**
  Open a window that shows all generated structures. Off as default.
- ▷ **Show tool bar**
  Shows the tool bar. On as default.
- ▷ **Show status bar**
  Shows the status bar. On as default.
- ▷ **Decoration**
  - ▷ **Mark starting node**
    Mark the starting node with a wave-shaped decoration. On as default.
  - ▷ **Mark expandable nodes**
    Mark nodes with small triangels at the top or bottom if they are expandable. On as default.
  - ▷ **Mark leaf nodes**
    Marke the leaf nodes, i.e., nodes that do not cover any new nodes.
- ▷ **Compact graph layout**
  Switch to compact graph layout. Off as default.
- ▷ **Zoom** Open the zoom dialog window.
- ▷ **Zoom to fit**
  Zoom the graph to fit the window.

▷ **Options**

- ▷ **Node distance**
  Open the dialog window to change the distance vertically and horizontally between the nodes in the graph.
- ▷ **Notation**
  - ▷ **Block structure notation**
    Change the structure notation to block structure notation.
  - ▷ **Weyr characteristics**
    Change the structure notation to Weyr characteristics.
  - ▷ **Segre characteristics**
    Change the structure notation to Segre characteristics.
- ▷ **Font size**
  - ▷ **12pt** Show structure information in 12pt font size.
  - ▷ **14pt** Show structure information in 14pt font size.
  - ▷ **16pt** Show structure information in 16pt font size.
- ▷ **Plug-in manager**
  Open the plug-in manager.
- ▷ **Save options**
  Save the options. This is automacically done when StratiGraph closes.

Button bar

Open the Graph Wizard.

Open a saved graph.

Save the current graph.

Export the graph to PostScript.

Expand the complete graph recursively.

Expand the active node upwards recursively.

Expand the active node downwards recursively.

Open a window that shows the closest neighbouring nodes to the active node.

Open a window with information of all the generated nodes.

Open the zoom dialog window.

Scale the graph to fit the window.

Exit StratiGraph.

APPENDIX C

Keyboard short cuts

Global short cuts

The action key is different on different systems. On a Windows based system it is usually the left CTRL key and on a UNIX based system it is usually the left Alt. key.

Action + O Open a saved graph.

Action + S Save the current graph.

Action + E Export the graph to PostScript.

Action + Q Exit StratiGraph.

Action + N Open the Graph Wizard.

Action + U Expand the active node upwards.

Action + Shift + U Expand the active node upwards recursively.

Action + D Expand the active node downwards.

Action + Shift + D Expand the active node downwards recursively.

Action + A Expand the complete graph recursively.

Graph window short cuts

U Expand active node upwards

D Expand active node downwards

↑ Move active node to the node with lowest order number on the nearest upper codimension level.

Move active node to the node with lowest order number on the nearest lower codimension level.

Move the active node to the node with nearest lower order number.

Move the active node to the node with nearest higher order number.

Bibliography

[1] J. Demmel and B. Kågström. Accurate solutions of ill-posed problems in control theory. *SIAM J. Matrix. Anal. Appl.*, 9(1):126–145, January 1988.

[2] J. Demmel and B. Kågström. The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications. Part I: Theory and algorithms. *ACM Trans. Math. Software*, Vol.19(No. 2):160–174, June 1993.

[3] J. Demmel and B. Kågström. The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications. Part II: Software and applications. *ACM Trans. Math. Software*, Vol.19(No. 2):175–201, June 1993.

[4] A. Edelman, E. Elmroth, and B. Kågström. A geometric approach to perturbation theory of matrices and matrix pencils. Part I: Versal deformations. *SIAM J. Matrix Anal. Appl.*, 18(3):653–692, 1997.

[5] A. Edelman, E. Elmroth, and B. Kågström. A geometric approach to perturbation theory of matrices and matrix pencils. Part II: A stratification-enhanced staircase algorithm. *SIAM J. Matrix Anal. Appl.*, 20(3):667–699, 1999.

[6] E. Elmroth, P. Johansson, and B. Kågström. Computation and presentation of graphs displaying closure hierarchies of Jordan and Kronecker structures. *Numerical Linear Algebra Applications*, 8:381–399, 2001.

[7] F. Gantmacher. *The Theory of Matrices, Vol. I and II (transl.)*. Chelsea, New York, 1959.

[8] B. Kågström. Singular Matrix Pencils. In Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. Van der Vorst, editors, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, pages 260–277. SIAM Publications, Philadelphia, 2000.

[9] B. Kågström and A. Ruhe. ALGORITHM 560: An algorithm for the numerical computation of the Jordan normal form of a complex matrix [F2]. *ACM Trans. Math. Software*, 6(3):437–443, 1980.

[10] B. Kågström and A. Ruhe. An algorithm for the numerical computation of the Jordan normal form of a complex matrix. *ACM Trans. Math. Software*, 6(3):389–419, 1980.

[11] P. Van Dooren. The generalized eigenstructure problem in linear system theory. *IEEE Trans. Autom. Contr.*, AC-26(1):111–129, 1981.

Index

A

algebraic multiplicity, **5**

B

block structure notation, **28**, 44
bundle, **8**, 9, 14, 18, 35–41

C

canonical structure transition, **23**, 37
closure, 1, **10**, **13**, 14, 15, 24
  hierarchy, **10**, 14, 15, 17
  relation, 3, 9, 15
codimension, 1, **9**, 10, 21, 22, 24, 41
compact layout, **30**
controllability, **7**, **39**
  pair, **7**, 8, 39
cover, **9**, 10, **13**, 14, 22, 23, 39

E

edge, **23**
eigenvalue, **5**, 6, 8, 9, 18, 28, 35, 39, 40
  coalesce, 37, 39
  defective, **5**
  derogatory, **5**
expansion, **21**, 22, 33, 35, 43, 47
  recursive, **22**, 23, 43, 45
export, **27**, 43, 45, 47

F

font size, **29**, 44

G

geometric multiplicity, **5**
graph wizard, **17–18**, 35, 43, 45, 47
GUPTRI form, **7**, 8, 11–14

I

input matrix, **7**
input vector, **7**, 41
integer partition, **9**

J

Jordan
  block, **6**, 7, 11–15, 18, 28, 29, 37, 41
  normal form, **6**, 7, 11
  structure, 1, **5**, 8–10

K

Kronecker
  canonical form, **6**, 7, 8
  structure, 1, **6**, 9, 10, 41

L

landscape format, **27**
linear state space model, **39**

M

manifold, 1, **9**
Matlab, **34**
matrix, **5**, 17, 18, 33
  diagonalizable, **5**
  nilpotent, **11**, 35, 36
  similar, **8**
matrix pair, **7**, 17, 18, 33, 42
matrix pencil, **6**, 7, 17, 18, 33, 37
  strictly equivalent, **8**
memory, 23, **33**
minimal indices, 6–8

N

node, **21**, 24

active, **22**, 24, 43, 45, 47
all generated, **24**
distance, **29**, 44
position, **24**
related, **24**
notation, **28**

O

observability, **7**, **39**
pair, **7**
open, **28**, 43, 45
orbit, **8**, 9, 14, 17, 35–40
order number, **21**
output matrix, **7**
output vector, **7**

P

plug-in, **30**, 34, 44
portrait format, **27**
PostScript, **27**, 33, 43, 45, 47
Encapsulated, **27**
principal vector, **5**, 12
problem setup, **17–18**, 30, 35
program extension, **30**

R

regular structure, **6**, 7, 37, 39, 40

S

save, **28**, 43, 45, 47
scale, **25**, 44, 45
Segre characteristics, **28**, 44
singular block, **6–7**, 18, 28, 35, 37
singular structure, **7**
staircase algorithm, 1, 11
state vector, **7**, 39, 41
status bar, 28, 44
stratification, 1, 7, **10**, 11, 23, 35–41
system matrix, **7**

T

tangent space, 9
tool bar, 25, 27, 44

U

uncontrollable mode, 41

W

Weyr characteristics, **12–14**, 28, 44